An Interop Publication

# CONNEXIONS®

## The Interoperability Report

July 1995 — *The 100th Issue* — Volume 9, No. 7

*ConneXions — The Interoperability Report tracks current and emerging standards and technologies within the computer and communications industry.*

## In this issue:

*ConneXions* is published monthly by Interop Company, a division of SOFTBANK Exposition and Conference Company, 303 Vintage Park Drive, Foster City, California, 94404–1138, USA.
Phone: +1 (415) 578-6900
Fax: +1 (415) 525-0194
E-mail: `connexions@interop.com`

*Subscription hotline:* 1-800-575-5717
*or* +1-502-493-3217

**ISSN 0894–5926**

### From the Editor

Here we are, 100 issues and some 3,000 pages later. It is tempting to look back and recount some of the moments from our past, but I will save all that for the 10th anniversary issue. Instead I would like to take this moment to thank all of the contributors to this journal. Without them there would be no *ConneXions.* The Editorial Board deserves a special note of appreciation for letting me know what I should focus on and for telling me when it's time to switch gears. Thanks also to the people at Globe Printing Company who transform a number of loose sheets of paper into a final bound "book" every month. Finally, a note of appreciation goes out to all our readers for giving us important feedback and suggestions. We continue to rely on your valuable input, which as always should be sent via e-mail to: `connexions@interop.com.`

Our "Back to Basics" series continues this month with a tutorial on AppleTalk. The technology surrounding the Apple Macintosh is near and dear to me since *ConneXions* is produced exclusively with Macintosh equipment. I have always been amazed at how simple the networking aspects of the Mac are. With the possible exception of MacTCP, everything is plug-and-play and very intuitive. The Internet suite of protocols could surely learn something from this elegant approach.

The phenomenal success of the Internet has led to the specter of address space depletion, an issue which is solved in the long term by the new version of IP—IPv6 as discussed several times in this journal. In the meantime, effective use of the existing (IPv4) address space is imperative. Yakov Rekhter and Tony Li discuss IP address allocation and management policies for the Internet in an article entitled "Address Ownership Considered Fatal."

Last month we published an article called "The Firewall Heresies." This month, Ted Doty brings you up to speed on this important area with a tutorial on basic firewall concepts.

In some parts of the world, Internet access is becoming an "off-the-shelf" item. Several complete solutions exist for a variety of computing platforms, all you need is a telephone line and a suitable modem, and you can be on-line within minutes. But this level of simplicity is by no means universal. Our user case study this month comes from Africa where attempts are underway to establish AAUnet, the African Universities Network. The article is by John Bart-Plange.

## *Back to Basics:* AppleTalk

**by Ed Tittel and Dave Smith**

### Users first

There's a scramble on in the computer world right now. Systems designers, developers, and even managers spend a lot of time talking about "end users." Graphical interfaces for applications, object-oriented operating systems, and graphical development environments give systems users more power than ever before to do what they want to do the way *they* want to do it.

It wasn't always this way. Computers are complex, technically oriented machines designed by complex, technically oriented people. For lots of reasons it always made more sense to computer system designers for users to adapt to the system than for the system to adapt to the users. In the beginning, system designers and implementors thought a lot about hardware and code and not so much about the people who had to use it.

Apple Computer, Inc. changed all that. Apple brought a new philosophy to the computer world—their computers were designed and built with users in mind. The company's stated philosophy is to provide greater power to individuals through computer technology. Before Apple, computers had been dedicated to a mission or to a business goal. Apple computers, represented by the Apple II and Macintosh family, have always been dedicated to their users, allowing direct manipulation of resources and capabilities.

### Let them talk

The designers at Apple realized early on that no person is an island. Even in the beginning, when they were making some of the first personal computers, Apple designers wanted to make "interpersonal" computers. In other words, they wanted users to maximize their own potential, and they wanted to allow interaction between users through their computers.

If the Macintosh is devoted to a user, then *AppleTalk,* the Apple networking protocol family, is dedicated to giving users access to resources and information outside the confines of their own desks, offices, or homes. As the Macintosh operating system was created to allow seamless control of a machine, the AppleTalk networking system was created to allow seamless access to other machines, devices, and the world.

### Why bother?

Apple's philosophy of giving users the power to make their own world carried over into their philosophy of how networking should take place. Back in the early 1980s, when AppleTalk was being created, the network world was a disorderly and confusing place.

### In the beginning

It had been hard enough to solve the big technical problems and create computers that worked reliably. It was easy for designers and builders of the first generations of computers to focus on the machines, and save the worry about the users of the machines for later.

Once the machines worked—more or less reliably—designers and builders turned their attention to networking. They brought with them their technically oriented, machine-centric philosophy. Networking technology was developed as a kind of adjunct, or add-on, to computer technology.

### The Apple touch

Apple engineers took a different approach—the Apple approach. They didn't want to give their users more complicated things to learn and worry about. So they decided to make networking technology an integral part of, and sometimes a seamless extension of, the basic Macintosh operating system.

They didn't want users to have to learn a machine operating system and then a network operating system. They wanted to create networking and internetworking with no extra baggage. They came to the conclusion that the network technology then in existence would not favorably extend their system, so they designed their own.

## Goals

AppleTalk's designers had a list of goals in mind when they created their protocol suite and its related services:

- *Versatility:* Design a network system that allows any machine or device to participate in the sharing of information.
- *Plug-and play capability:* Always an Apple specialty network devices should be plugged in to a network and begin to operate immediately with no special configuration necessary.
- *Peer-to-peer architecture:* Save money and increase flexibility by making all network machines equals. Save the money for dedicated file servers or devices, and spread the processing load over as wide a net as possible.
- *Simplicity:* The simpler the network operating system is, the less overhead necessary to implement it. With less overhead comes the ability to build network capability right into the machine operating system itself.
- *Link independence:* Since technology will advance and take unforeseen directions, network architectures cannot be too closely linked to current hardware and software.
- *Seamless extension of the user's computer:* Maintain the look and feel of the individual machine's operating system throughout the network operating system.
- *Open architecture:* Allow users and third party vendors to add to and extend the operating system in ways not originally contemplated.

## The Apple stack

AppleTalk's architecture corresponds closely to the OSI model for network operating systems. Let's take a look at the AppleTalk model (Figure 1), then we'll see how it "stacks" up against the OSI model.

This is a graphic representation of the AppleTalk protocol stack. Like all protocol stacks, each layer operates independent of the other layers, providing for ease of maintenance and flexibility in implementation.

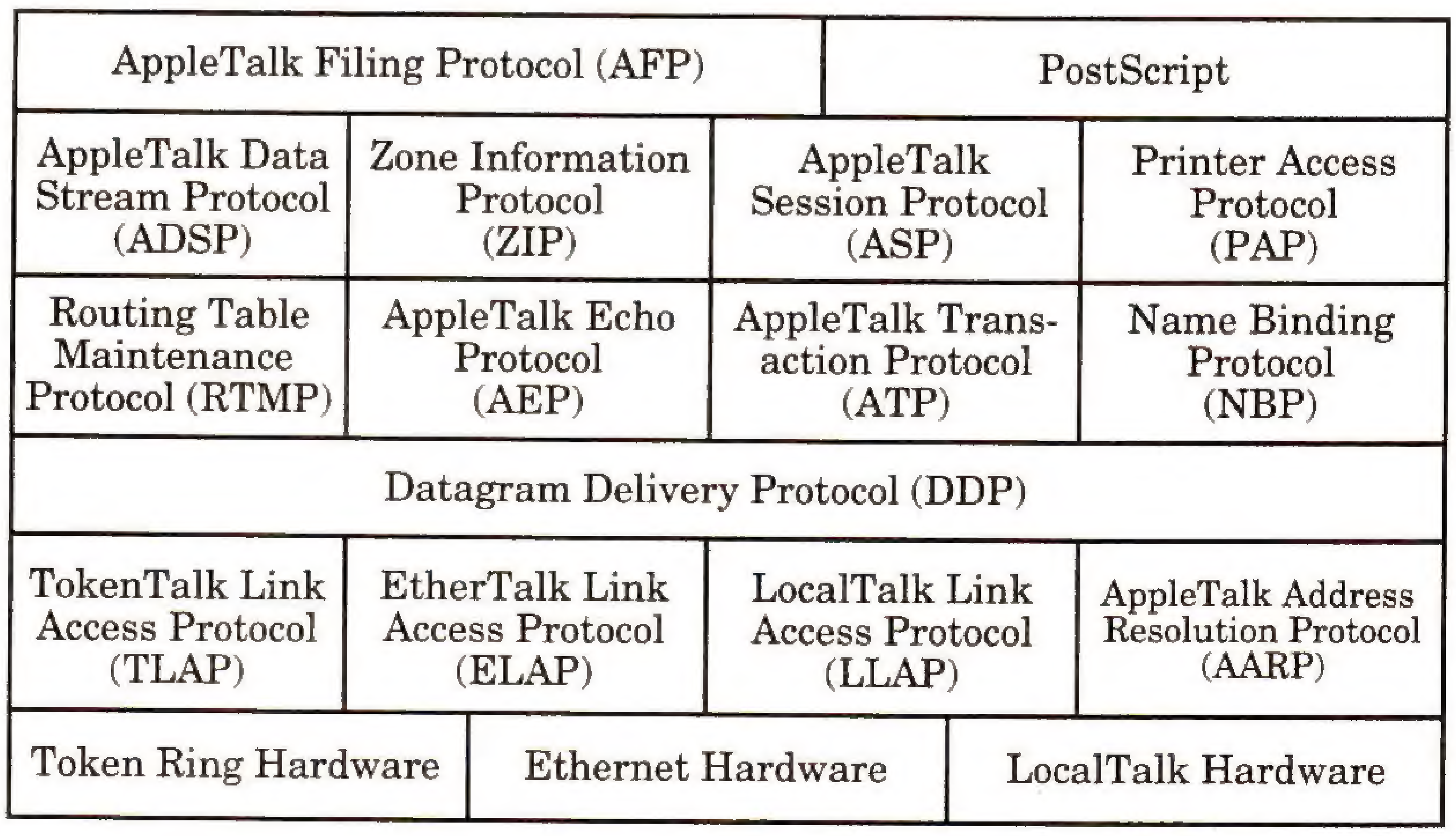
AppleTalk Filing Protocol (AFP)
PostScript
AppleTalk Data Stream Protocol (ADSP)
Zone Information Protocol (ZIP)
AppleTalk Session Protocol (ASP)
Printer Access Protocol (PAP)
Routing Table Maintenance Protocol (RTMP)
AppleTalk Echo Protocol (AEP)
AppleTalk Transaction Protocol (ATP)
Name Binding Protocol (NBP)
Datagram Delivery Protocol (DDP)
TokenTalk Link Access Protocol (TLAP)
EtherTalk Link Access Protocol (ELAP)
LocalTalk Link Access Protocol (LLAP)
AppleTalk Address Resolution Protocol (AARP)
Token Ring Hardware
Ethernet Hardware
LocalTalk Hardware

Figure 1: The AppleTalk stack

*continued on next page*

## AppleTalk *(continued)*

Figure 2 maps the AppleTalk protocol stack to the OSI reference model. You might note that AppleTalk's architecture very closely resembles the well-known OSI Reference Model.

<table>
<tr><th>OSI</th><th colspan="12">AppleTalk</th></tr>
<tr><td>Application</td><td colspan="6">AppleTalk Filing Protocol (AFP)</td><td colspan="6">PostScript</td></tr>
<tr><td>Presentation</td><td colspan="12"></td></tr>
<tr><td>Session</td><td colspan="3">Routing Table Maintenance Protocol (RTMP)</td><td colspan="3">AppleTalk Echo Protocol (AEP)</td><td colspan="3">AppleTalk Transaction Protocol (ATP)</td><td colspan="3">Name Binding Protocol (NBP)</td></tr>
<tr><td>Transport</td><td colspan="3">Routing Table Maintenance Protocol (RTMP)</td><td colspan="3">AppleTalk Echo Protocol (AEP)</td><td colspan="3">AppleTalk Transaction Protocol (ATP)</td><td colspan="3">Name Binding Protocol (NBP)</td></tr>
<tr><td>Network</td><td colspan="12">Datagram Delivery Protocol (DDP)</td></tr>
<tr><td>Data Link</td><td colspan="4">TokenTalk Link Access Protocol (TLAP)</td><td colspan="4">EtherTalk Link Access Protocol (ELAP)</td><td colspan="4">LocalTalk Link Access Protocol (LLAP)</td></tr>
<tr><td>Physical</td><td colspan="4">Token Ring Hardware</td><td colspan="4">Ethernet Hardware</td><td colspan="4">LocalTalk Hardware</td></tr>
</table>

Figure 2: Mapping AppleTalk onto the OSI Reference Model

Since they were starting from scratch to create a user friendly, plug-and-play network architecture, the developers of AppleTalk were able to build discrete, well-defined protocols to operate at each layer of their stack. Let's take a brief look at each layer of the OSI stack and the protocols that operate at the corresponding layer of the AppleTalk stack.

### Physical layer

As we've said before, the physical layer is defined by the type of network interface card and the medium (type of wire or cable) used to physically connect the machines operating on a network. Network interface cards are usually designed to operate with one particular access method. Because different networking systems are in operation and therefore different hardware in use, AppleTalk's designers created protocols that could support a variety of methods, including Ethernet, Token Ring, ARCnet, FDDI, and Apple's own LocalTalk (a 230 Kbps twisted-pair networking technology).

Let's move on to the data link layer, since it operates so much in conjunction with the physical layer, and we'll take a look at some of this variety.

### Data Link layer

Every Apple computer made comes with the physical connection hardware and the protocols to transfer information from a machine or from the higher layers of the stack onto the physical transport medium.

The hardware, *LocalTalk* hardware, and the *LocalTalk Link Access Protocol* (LLAP) operating at the data link layer are built into every Apple computer. Taken together, these are enough to enable real peer-to-peer networking. All you need is some cable to connect the machines.

AppleTalk's creators knew that some popular networking access methods and hardware were already in use. That's why their protocol stack provides support for many of the most popular ones: Token Ring, ARCnet, FDDI, and Ethernet, as well as LocalTalk.

At the data link layer the *TokenTalk Link Access Protocol* (TLAP) and *EtherTalk Link Access Protocol* (ELAP) provide the translation between the higher layers of the AppleTalk stack and their corresponding supported physical link. Similar data link protocols operate for the other networking technologies.

Also operating at this layer is the *AppleTalk Address Resolution Protocol* (AARP). AARP is responsible for translating protocol addresses from higher up in the stack into hardware addresses.

**Network layer**

This layer of the OSI stack corresponds to the AppleTalk layer that features the *Datagram Delivery Protocol* (DDP), the workhorse protocol of the AppleTalk suite. Data link layer protocols provide for the delivery of information between different stations, or nodes, on a LocalTalk network. DDP extends this capability by providing for delivery of information between internet sockets—higher layer process addresses. We'll discuss how AppleTalk defines a socket later on in this article.

Like its counterparts in other networking stacks, DDP is a connectionless, best-effort protocol. It handles the necessary segmentation and addressing of packets to be transported, and lets other protocols worry about error control and transmission reliability.

**Transport layer**

This layer of the OSI stack holds the protocols that are the key to the operation of an AppleTalk internetwork, in that they provide the key services needed by DDP for locating, addressing, and routing transmissions on the internet.

The *Routing Table Maintenance Protocol* (RTMP) maintains, updates, and broadcasts internet addresses and routes. RTMP operates on an AppleTalk internet router, maintaining and providing information about destination addresses on the internet.

The *AppleTalk Echo Protocol* (AEP) operates out of a designated socket on each node in an AppleTalk internetwork. AEP finds out if a particular node is accessible on the internet and finds out the time for round-trip data transfer to a particular node.

The *AppleTalk Transaction Protocol* (ATP) provides reliable transport services between sockets on a node. AppleTalk defines a transaction as a request from one socket to another to perform a high layer function and report on the status of the requested service.

The *Name Binding Protocol* (NBP) is the protocol that provides for resolution of the names of devices and machines on the AppleTalk internetwork. NBP translates internet numerical addresses into the actual names and aliases of machines and devices. An AppleTalk internetwork has no designated name server. NBP takes care of maintaining information about all reachable names and addresses.

**Session layer**

At the session layer of the OSI stack, the AppleTalk developers were determined to provide the flexibility that is uncommon in network architectures by providing session protocols tailored to very specific needs as well as general ones.

The *Zone Information Protocol* (ZIP) allies itself with the routing protocols in operation at lower levels to allow networks and internetworks to be broken down into less unwieldy partitions. We'll define what zones mean in an AppleTalk internetwork later on.

The *AppleTalk Data Stream Protocol* (ADSP) is designed to provide general session layer services between any two sockets on an internetwork. ADSP establishes and maintains data stream transmissions between sockets with full duplex capability.

The *AppleTalk Session Protocol* (ASP) was included specifically to be used by ATP to establish two-way communication between workstations and servers on an AppleTalk internet.

*continued on next page*

## AppleTalk *(continued)*

The *Printer Access Protocol* (PAP), also allied closely with ATP, was originally included in the stack to handle communications between Apple computers and LaserWriter printers. Today, PAP is used to provide a connection-oriented communication between clients and servers, as well as between nodes and printers. Like ASP, PAP uses ATP and NBP to establish and maintain connections between clients and servers on the internetwork.

### Presentation and Application layers

In the AppleTalk stack, the line between the presentation and application layers of the OSI stack is somewhat hazy. Two protocols are operating at this layer, the *AppleTalk Filing Protocol* (AFP) and *PostScript.*

AFP allows workstation users to share files by providing information on the location of a file on the network and a means of accessing the file. *PostScript* is a particular page description language developed by Adobe Systems, used for transferring information between Apple computers and *PostScript*-capable printers (like LaserWriters) on the network.

### LocalTalk (data link and physical layers)

As we've already mentioned, all Apple computers are delivered to users network-ready. What this means is that the networking protocols necessary for peer-to-peer networking are included in the basic Macintosh operating system. The necessary physical links are provided for by including a network interface built into each machine.

In fact, all you need to set up a LAN with Apple products is some cables to hook all the computers and devices together. Even LaserWriter printers are shipped network-ready. This readiness to network is important evidence of the goals that motivated the AppleTalk developers. In most ways that count, LocalTalk networks really are an extension of the Macintosh operating system.

The capability to just hook machines and devices together and provide automatic network capability is very important to Apple. Very little special configuration is necessary to install or configure a LocalTalk network. This is the embodiment of Apple's "plug and play" philosophy.

An Apple network that is not connected to another network works just fine with the basic protocols and physical connection media included with each machine and device. Apple, Inc. has indeed provided all the physical layer and data link layer protocols necessary for simple networking. Today, its more recent Quadra and PowerPC models come equipped with Ethernet, perhaps in recognition that LocalTalk's 230 Kbps bandwidth is inadequate for higher end network access.

### LLAP

The *LocalTalk Link Access Protocol* (LLAP) headers include the addresses of the source and destination nodes and a *type* field that specifies the packet type contained in the packet's data field. A node's address is dynamically assigned by LLAP when it starts up.

Different specifications operate for data packets and control packets. After the packet type comes a data length field followed by the data itself. A frame trailer follows with a *Frame Check Sequence* (FCS) number, a flag value, and an abort sequence indicating the end of the frame.

Figure 3 gives us a look at the internals of an LLAP frame.

| Frame Preamble | Destination Node ID | Source ID | LLAP Type | Data Length | Data | FCS | FLAG | Abort |
|---|---|---|---|---|---|---|---|---|

Figure 3: The layout of an LLAP frame

Control packets are used to provide information to the LAN and do not contain data. Data packets can contain up to 600 bytes (called *octets,* in network-speak for some reason lost in the dim history of data communications) of information.

### ELAP and TLAP

AppleTalk developers designed support for other popular networking methods into their protocol architecture. This includes the *Ethernet Link Access Protocol* (ELAP) and the *Token Ring Link Access Protocol* (TLAP), as well as other link access protocols.

The EtherTalk Link Access Protocol encapsulates the information contained in an LLAP frame in an Ethernet frame by adding an Ethernet destination source field, Ethernet source field, and Ethernet type field. Some padding may be added to reach the minimum length requirements for Ethernet frames. The TokenTalk Link Access Protocol operates exactly the same way. A Token Ring destination field and a source field are added to the LocalTalk header.

### AARP

The *AppleTalk Address Resolution Protocol* (AARP) is implemented at the data link layer to provide a way to handle transmissions between the higher level protocols and hardware addresses. AARP uses three types of packets to do its job: *probe, request,* and *response* packets. Here's how they work.

When a network node initializes AARP assigns an address for the higher level protocol stack on the node and then broadcasts probe packets to find out if any other nodes are using that address.

Each node keeps an address mapping table for any protocol stacks on the device. Request packets are sent out to locate specific nodes on the link so that a transmission can occur. If a node gets a request for a protocol address that matches its own it sends out a response packet containing its hardware address. If a match is not found then a request packet is sent out.

Every protocol suite involved in an internetwork needs a connectionless protocol in operation at the network layer to provide for delivery of packets. While LAN transmissions are enabled by data link and physical layer protocols, the protocols at the network layer make internetworking possible.

### DDP

In fact, all the AppleTalk protocols in operation above the *Datagram Delivery Protocol* (DDP) layer depend on DDP to let them do their work. The data link layer frames we've already discussed deliver information between nodes on a LAN. DDP is responsible for delivering packets between sockets on the internet. A *socket* is defined as the addresses of higher layer processes on any given network node.

### Sockets explained

All high level processes operating at protocol stack layers above DDP are dynamically assigned socket numbers by DDP, or they have statically assigned sockets reserved for use by protocols such as the Name Binding Protocol and Routing Rable Maintenance Protocol. Internet socket addresses are made up of a 16-byte network number, an 8-byte node ID, and an 8-byte socket number.

*continued on next page*

## AppleTalk *(continued)*

DDP gets a node's AppleTalk address when it initializes. DDP takes information from the underlying data link in use by the node, and from internet routers on the data link, to get this address.

Socket clients are implemented with socket listeners that receive datagrams addressed to that particular socket. Socket listeners are able to receive asynchronously either through an interrupt or some type of input/output routing. A sockets table is implemented in DDP to maintain the location and type of each open sockets listener. Calls to the socket listener are used to open a dynamically assigned socket, close a socket, send datagrams, and receive datagrams.

### DDP in operation

Datagrams are transmitted by internet routers from source to destination sockets. DDP on the source node checks the destination network number of the datagram and finds out if the destination is on the local network or somewhere beyond the local link. If the destination node is on the LAN, then DDP hands the packets to be sent down to the data link layer for transmission. If the destination is not local, then DDP encapsulates the transmission in its header and uses the data link protocols to send the packet to a router. The router uses its routing tables and the associated routing protocol to determine the best path to the destination and forwards it onto the internetwork. A router on the destination network is eventually reached and hands the datagram down to the data link layer for forwarding to the destination node.

### DDP packets

DDP packets are made up of an 8-bit hop count field, 8-bit datagram length field, 16-bit DDP checksum, 16-bit destination network number, 16-bit source network number, 8-bit destination node ID, 8-bit source node ID, 8-bit destination socket number, 8-bit source socket number, and an 8-bit DDP packet type field encapsulated in a LLAP frame along with the accompanying data. Figure 4 shows an example of the formation of a DDP packet. It is a representation of the makeup of an DDP packet header.

| | |
|---|---|
| Hop count | (1 byte) |
| Datagram length | (1 byte) |
| DDP checksum | (2 bytes) |
| Destination network | (2 bytes) |
| Source network | (2 bytes) |
| Destination node | (1 byte) |
| Source node | (1 byte) |
| Destination socket | (1 byte) |
| Source socket | (1 byte) |
| DDP Type | (1 byte) |
| DATA | (0 to 586 bytes) |

Figure 4: The anatomy of a DDP packet header

### Routing tables

Routers need to know the addresses of all other available routers and therefore about all the other active networks reachable on its internet. AppleTalk routers keep a complete listing of these networks, listed by their network numbers. This is known as a *routing table.* Each router in an AppleTalk internetwork will contain a routing table. The entries in the routing table let the router determine which is the best path for forwarding packets.

A routing table contains a listing of network numbers and an associated path used by DDP to deliver packets to their final destination network. The entries in the routing table contain no physical addresses of the network stations that reside on the internet.

The only physical addresses in the table are those of other routers to which packets, destined for a remote network, may be addressed. Routers do not know which other end stations are on the networks they connect to. The final destination (physical address of the final destination) is embedded in the DDP header. Their job is to get the packets to the right network, not to the precise workstation (this is handled at the local level by the final router in the chain between sender and receiver).

**RTMP**

The *Routing Table Maintenance Protocol* (RTMP) operates on AppleTalk internet routers. RTMP establishes and maintains the routing tables used to transmit datagrams over the internet between source and destination sockets. Every AppleTalk router has RTMP in operation on a statically assigned socket known as the RTMP socket. Routers get RTMP packets from other routers on the internet and use the information they contain to maintain and update their own routing tables.

Each entry in the routing table includes the port number through which packets must be forwarded by the router, the network number and the node address of the next router, and the distance to the destination network measured in *hops*.

RTMP provides for the construction of routing tables and keeps track of other routers initializing or going down on the internet. When a router initializes it starts its own table by examining each of its enabled ports. Any port with a network number other than zero connected to one of the router's ports tells the router that it is directly connected to that network. RTMP creates table entries for these network numbers.

Each router in the internetwork broadcasts its own table periodically (every 10 seconds) by sending data packets through its enabled ports to the RTMP socket. If a router receives RTMP table information not included in its own table, then that information is added to its own table. In this way, information about a network's configuration gets updated regularly and reliably.

**AEP**

The *AppleTalk Echo Protocol* (AEP) is used to find out if a node is accessible over the internet. AEP operates on a statically assigned socket and listens for packets addressed to this socket. If a packet is received the echoer determines if the packet is an echo packet. If it is, then a copy of the packet is made and transmitted to the same socket on the sending node.

AEP is used to determine if nodes can be reached and to measure the round-trip time for a packet to reach a remote node and be returned (much like Ping in the TCP/IP stack). This information is valuable for certain proactive network management functions and is used by higher level protocols in estimating the time-outs specified by protocols at higher levels in the stack.

**NBP**

AppleTalk protocols rely on *numbers*, node ID numbers, socket numbers, network numbers, etc. to provide addressing capability. But, as we all know, numbers, which are easy for computers to remember, are not so easy for users to remember. Names are a more usable type of identification for the average user. When a user refers to a network device or computer by name, that name must be converted into a network address that can be understood by the computers on the internetwork. The *Name Binding Protocol* (NBP) handles the translation of numbers into names, and vice versa.

*continued on next page*

## AppleTalk *(continued)*

Since node address assignments are done dynamically by AppleTalk, the number assigned to a node may change often. The name of that particular node does not change so readily. NBP is valuable in keeping up with dynamically assigned names and addresses.

**NVEs**

Each node on the network maintains a names table that keeps the names to entity internet address mappings of all the *Network Visible Entities* (NVEs) in that node. An NVE is any entity accessible to DDP on an AppleTalk network. Nodes on the internet are not NVEs but the services in those particular nodes are. A network server is not an NVE, but the services offered on defined sockets in that server are NVEs.

The NBP process is statically assigned to a socket and responsible for keeping the node's names table and handling lookup requests from the network and from the node itself.

**NBP services**

NBP provides four basic services:

- Name registration
- Name deletion
- Name lookup
- Name confirmation

NVEs enter their names and socket numbers into the names table by notifying NBP on the node. NBP is responsible for entering these names into the table. If an entity stops operation for some reason, it sends a name deletion call to NBP and the corresponding mapping is removed from the names table.

When a user or higher level protocol wants access to an entities service, it issues a name lookup call to NBP. NBP performs a search through the names table and returns the address to the requester. In some circumstances, NBP issues a name confirmation call to confirm that a mapping is still valid.

**ATP**

The *AppleTalk Transaction Protocol* (ATP), operating in the transport layer of the stack, is concerned with end-to-end data flow between sockets on the internetwork Earlier, we described DDP, working at the network layer, as a best-effort or connectionless protocol. That means DDP is interested only in forming up a packet, addressing it, and sending it on its way.

ATP, being concerned with end-to-end data flow, is connection oriented. That means ATP wants to make sure that all its packets are delivered error free. Also, ATP is concerned with the state of the connection and the results of a transmission.

**Transactions**

ATP is designed to create and maintain an error free, transaction-based, connection between sockets. When a socket client on an AppleTalk internet requests a service from another socket client, it sends a request to that socket using ATP. The socket client receiving the request performs the service as requested and then sends an ATP response out to the requester reporting the outcome. This interaction between the requester and service performer is called a *transaction.* ATP was designed specifically to take care of this type of network activity.

Two types of transactions are the responsibility of ATP:

- At least once (ALO)
- Exactly once (XO)

When a socket client on a network node sends out a request for services it makes the determination whether the requested transaction is an ALO or an XO transaction.

ALO transactions make sure that the requested service is performed at least once, and ATP ALO is concerned if the service is performed more than once. If the requesting socket doesn't receive an ATP response from the service provider, it just sends out another request. If for some reason the response packets aren't getting through, then the transaction requested is performed each time a request comes through. Asking a remote node for its ID is an example of an ATP ALO transaction.

XO transactions can't be executed more than once without causing some type of unwanted change. Some protocols higher up in the AppleTalk stack, like the printer access protocol and AppleTalk session protocol require this type of service to maintain the integrity of their transmissions. ATP XO maintains a list of requests and corresponding responses. When a request for a transaction is sent out with ATP XO, the receiving socket checks the list to see if the transaction has been received and responded to already. If this is the case, ATP XO ignores the request.

**Session layer**

On the next level of the stack, the sessions layer, AppleTalk provides four protocols concerned with establishing and maintaining connection-oriented reliable data transfer: PAP, ASP, ADSP, and ZIP. For more details on these protocols, please read on.

**PAP**

The *Printer Access Protocol* (PAP) is a session layer protocol that sets up connection-oriented communications sessions between workstation and server socket clients on the AppleTalk network. The protocol was originally designed strictly for workstation communication with LaserWriter and ImageWriter printers but it can be used for any asymmetrical communications between servers and workstations.

PAP defines a node as having multiple processes that are available to workstations on the internetwork. These processes are made available by the server to the client workstation through PAP, which is itself a client of ATP and NBP. Servers advertise their availability to the network through session listening sockets.

When a workstation wants to establish a connection with a server node, it will call NBP for the address of the server's session listening socket, then call the address to request a connection session. PAP uses ATP, in exactly one mode, for the data transmission.

PAP servers can handle many transactions simultaneously, depending on their configuration. If all available connections are in use on the server, it will ignore other PAP requests for session connections.

**ASP**

The *AppleTalk Session Protocol* (ASP) opens up sessions between sockets on the internet and is responsible for maintaining the connection and closing down the session once the transmission is complete. ASP uses ATP as a delivery mechanism to perform four basic functions related to communication between the workstations requesting services and servers provided the requested services.

*continued on next page*

## AppleTalk *(continued)*

The four basic functions are:

- Opening sessions
- Session request handling
- Session management
- Closing sessions

ASP requests between clients are sent out to open a session. Session request handling describes the transfer of messages and data between the sockets involved in the session. Session management is concerned with the integrity of the data flow and the status of the sockets involved in the session. At the end of a session, ASP closes down the connection.

When ASP initiates a session or connection between two network entities, it assigns a session number to the connection, so that more than one workstation may access the server at the same time. ASP commands are sent to the server and ASP replies are returned to the workstation. ASP does not provide for the server to make demands on the workstation, but it does allow the server clients to let the workstation know when it needs attention.

ASP's reason to exist is to establish and maintain direct connections between the higher level protocols in the stack with a minimum of alteration by the lower level protocols. A server entity on the network makes its service known by opening an ATP socket and then registering a unique name on this socket with the Name Binding Protocol. Once the name is registered, ASP is notified of the address on the ATP socket. ASP listens at this socket for session requests from workstations that want to use the advertised service. Workstations call NBP to learn the socket's address and then call ASP to open the session.

### ADSP

The *AppleTalk Data Stream Protocol* (ADSP) takes the functionality of ASP a little bit further. ADSP is a connection-oriented protocol that allows client sockets to establish and maintain a two-way byte stream of data. ADSP concerns itself not only with the reliability of the transmission but also the flow control and sequencing of bytes by assigning sequence numbers to the bytes in the stream. With ADSP, the flow of data between two socket clients is directed to and from the connected nodes simultaneously.

### Making the connection

A connection end of ADSP communication is identified by its internet socket address, made up of the socket number, the node ID, and the network number. ADSP assigns a connection ID to each session so that each sequenced byte transmitted can be forwarded to the right process on the right node at the right time.

### Byte numbers

ADSP also assigns a sequence number to each byte in the data stream between the two connection ends in a network transmission. Each packet transmitted carries a sequence number that ADSP uses to maintain error-free delivery of the packet flow.

### Flow control

ADSP also uses the sequence numbers to maintain flow control, so that slow receivers aren't flooded with more packets than they can deal with. Periodically, ADSP checks the receive buffers on the node to make sure that they aren't filling up with data. Information about the buffer status is exchanged between the ADSP socket clients.

### ZIP

AppleTalk's designers built AppleTalk with the ability to service up to 16 million nodes on a single internet. An internet is made up, of course, of a number of local area networks connected by routers to form an internet. With AppleTalk, the nodes on an internet can be segmented into *zones,* allowing the administrators to exercise additional control. There is no strict relationship between zones and network numbers on an internet. Nodes on the same network could be in different zones, at the discretion of an administrator with sufficient privileges to assign and modify AppleTalk zones.

Like RTMP, the *Zone Information Protocol* (ZIP) operates on AppleTalk internet routers. NBP makes calls to ZIP on the router to find out what networks or network nodes belong in what zone. ZIP maintains the zone information in zone information tables maintained on the router. Zone information is transmitted to other routers on the internet through a statically assigned socket known as the *Zone Information Socket.*

### Application layer services

At the top of the stack, AppleTalk provides some specific services to users of the AppleTalk network. Two distinct protocols operate more or less along the border between the presentation and application layers of the stack. *PostScript,* the language used for Apple printer communications, and the AppleTalk File Protocol.

### AFP

The *AppleTalk File Protocol* (AFP) is a presentation layer protocol, a client of ASP. AFP is used to manipulate files on remote workstations and servers. AFP contains translators that can translate file formats between end-user nods on the internet.

Workstation clients request and manipulate files using the workstation's native file system commands. The native file system manipulates files on physically connected resources. A file data structure in local memory on the workstation indicates whether the requested file is local or if it resides in an external file system. If the file is local it is handled by the native file system. If the file is on an external file system, the native file system routes the command to an AFP translator.

The AFP translator translates native commands into AFP and sends them to the file server where the requested file resides. The translated commands are forwarded through the AFP AppleTalk filing interface.

### PostScript

*PostScript* is the presentation language used by AppleTalk stack when printer data is transferred from workstation clients to servers on the internet. *PostScript* is designed to be carried by ASP.

### Finishing the core

This article has presented the basic operating concepts of an AppleTalk internetwork. AppleTalk's designers decided early on in network history to design and build their own protocol stack. They succeeded in creating an elegant set of networking protocols that met the goals of creating a plug-and-play internetwork requiring an minimum of user configuration and operating as a natural extension of the native operating system. Even today, it remains among the simplest and easiest of networking technologies to use.

*continued on next page*

## AppleTalk *(continued)*

### Annotated bibliography

[1] Andrews, Richard F., Oppenheimer, Alan B., Sidhu, Gursharan S., *Inside AppleTalk,* 2nd ed., Addison-Wesley, 1990, Reading, MA. *This is the definitive work on AppleTalk, written by the Apple guys who helped design and create the AppleTalk protocols.*

[2] Apple Computer, Inc., *AppleTalk Network System Overview,* Addison-Wesley, 1989, Reading, MA. *Apple's own introductory overview of Macintosh networking and AppleTalk technologies (dated, but still useful).*

[3] Apple Computer, Inc., *Planning and Managing AppleTalk Networks,* Addison-Wesley, 1991, Reading, MA. *A book aimed at helping network administrators plan, install, and manage AppleTalk networks (dated, but also still useful).*

[4] Howard, Stephen. "AppleTalk: Old Protocol Poised for Speed, Mobility," *MacWEEK,* Volume 8, No. 1 (January 3, 1994), page 82.

[5] Kosiur, Dave and Joel Snyder, *The Macworld Networking Bible,* 2nd ed., IDG Books Worldwide, 1994 Indianapolis. *An outstanding comprehensive treatment of AppleTalk networking, from protocols, to hardware, to software, to troubleshooting and installation.*

[6] Miller, Mark A., *LAN Protocol Handbook,* M&T Publishing, Inc., 1990, Redwood City, CA. *An introductory look at AppleTalk, not a bad place to start.*

[7] Minshall, Greg, "AppleTalk versus IP," *ConneXions,* Volume 3, No. 9, September 1989. *Interesting discussion of the pros and cons of both IP and AppleTalk.*

[8] Woodcock, Bill, *Networking the Macintosh: A Step-by-Step Guide to Using AppleTalk in Business Environments,* McGraw-Hill, Inc., 1993. *A practical guide to planning, constructing and managing Macintosh networks.*

**ED TITTEL** is a full-time freelance writer and networking consultant, and a member of the NetWorld+Interop Program Committee. Ed is the author of ten books on computer topics, many of them network-related. In addition to *NetWare for Dummies,* now in its 2nd edition, Ed is the author of the forthcoming IDG books *HTML for Dummies* and *Foundations of WWW and HTML Programming,* as well as books on the Internet, network design, and e-mail for Academic Press. In his spare time, Ed likes to walk his hefty but personable Labrador, Dusty, and to share the blessings of domesticity with his wife, Suzy, and stepchildren, Austin and Chelsea. His e-mail address is: `ed@etittel.zilker.net`

**DAVE SMITH** is a network industry consultant and freelance writer, who has labored in obscurity under the corporate umbrella until quite recently. A collaborator with Ed on the forthcoming Academic Press Professional book: *The PC Networking Handbook* (from which this article has been adapted), Dave is also the author of numerous WWW pages and a variety of marketing-oriented high-tech literature. A denizen of Dripping Springs, Texas, Dave enjoys country livin' and his family from the vantage point of a lovely piece of property in the beautiful Hill Country. His e-mail address is: `dbsmith@comland.com`

[This article is based on material in *The PC Networking Handbook,* by Ed Tittel and Dave Smith, ISBN 0-12-691398-6, due to be published this month by Academic Press Professional. Used with permission. —Ed.]

# Address Ownership Considered Fatal

by
Yakov Rekhter and Tony Li, cisco Systems

> "Ah, I love these addressing and routing debates. After this I'm going to go have a root canal, no two root canals done, for more fun."
>
> *—Noel Chiappa*
> *(from his message to the IETF mailing list on 10/31/1992)*

## Introduction

IP address allocation and management are essential operational functions for the Internet. The exact policies for IP address allocation and management continue to be the subject of many discussions. Such discussions cannot be pursued in a vacuum—the participants must understand the technical issues and implications associated with various address allocation and management policies.

In this article we articulate certain relevant fundamental technical issues which must be considered in formulating IP address allocation and management policies for the Internet. The major focus of this article is on one possible policy, the concept of "address ownership," and the technical implications of this concept for the Internet.

## Address allocation and management policies

IP address allocation and management policy is a fairly complex, multifaceted issue. It covers a broad range of issues, such as who formulates the policies, who executes the policies, what are the roles of various Internet registries (e.g., The InterNIC, The RIPE NCC, etc...), what are the roles of various organizations (e.g., The Internet Society (ISOC), The Internet Architecture Board (IAB), The Internet Engineering Steering Group (IESG), The Internet Engineering Task Force (IETF), The Internet Engineering Planning Group (IEPG), various government bodies, etc.), the participation of end users in requesting addresses, and so on.

## Address ownership

Support for address ownership is one possible address allocation and management policy. Supporting address ownership means that once a block of IP addresses is allocated or assigned to an organization, the organization may always use these addresses. This implies that the organization will never need to change addresses (renumber), even if the organization changes its interconnection to the Internet (e.g., the organization changes its Internet Service Provider).

Some parties have asserted that the Internet assumes the concept of address ownership already. At the same time the fact remains that the concept of address ownership has never been explicit—the words that have been used have been "assignment," "allocation," and "delegation." One should also note that even if the current allocation procedures may be viewed as consistent with "address ownership," this does not mean that address ownership is, in fact, the current policy.

Address allocation and management and the ability of the Internet routing system to scale are not independent—only certain address allocation and management policies yield scalable routing. Since address ownership is a significant factor in such policies, we need to understand its implications on the scalability of the Internet routing system.

## Relationship between addressing and routing

In the abstract one could think of IP unicast addresses as just the set of integers in the range `0x01000000 - 0xdfffffff` (or using the traditional IP notation `1.0.0.0` through `223.255.255.255`).

*continued on next page*

## Address Ownership Considered Fatal *(continued)*

What makes an IP address unique from a practical perspective is its ability to gain access to the Internet routing service and thereby exchange data with the remainder of the Internet. IP addresses are used for Network Layer (IP) routing, thus an IP address of a node (e.g., a host) is the sole piece of information about the node that is injected into the routing system. In other words, it is the reachability of an IP address that gives it an intrinsic value—without reachability the address is worthless.

The above implies that it is the service environment—the Internet, and its continued operation, including its routing system, which provides an IP address with its intrinsic value, rather than the inverse. Consequently, if the Internet routing system ceases to be operational, the service disappears, and the addresses cease to have any functional value. At this point in the context of the Internet, all address allocation and management policies, including address ownership, are rendered meaningless.

### Hierarchical addressing in the Internet

*Classless Inter-Domain Routing* (CIDR) [2], [3] is presently used in the Internet as the primary mechanism to contain the growth of routing information—without CIDR the Internet routing system would have already have collapsed.

CIDR is based on the technique of *hierarchical routing* [1]. Hierarchical routing works by taking a set of addresses and generating a single routing advertisement for the entire set. Further, if addresses are assigned correctly, this can be done recursively: multiple advertisements can be combined into a single advertisement. By exercising this recursion, the amount of information necessary to provide routing can be decreased substantially.

A common example of hierarchical routing is the phone network (and more precisely, the *North American Numbering Plan*), where country codes, area codes, exchanges, and finally subscriber lines are different levels in the hierarchy. In the phone network, a switch need not keep detailed routing information about every possible subscriber in a distant area code. Instead, the switch knows one routing entry about the entire area code.

### Scaling

Notice that the effect on scaling is dramatic. If we look at the space complexity of the different schemes, the switch which knows about every subscriber in the world needs $O(n)$ space for $n$ worldwide subscribers. Now consider the case of hierarchical routing. If we break $n$ down into the number of subscribers in the local area ($l$), the number of other exchanges in the area code ($e$), the number of other area codes in the local country code ($a$) and the number other country codes ($c$), then hierarchical routing has space complexity $O(l+e+a+c)$. Notice that each of these factors is much, much less than $n$, and grows very slowly, if at all. This implies that a phone switch can be built today which has some hope of not running out of space as soon as it is deployed.

The fundamental property of hierarchical routing that makes this scalability possible is the ability to form abstractions: in this case the ability to group subscribers into exchanges, area codes and country codes. Further, such abstractions must provide useful information for the ability to perform routing.

Some abstractions, such as the group of users with green phones, are not useful when it comes time to route a call. Since the information that routing really needs is the location of the address within the topology, it is essential that an address in the network reflect its topological location within the network. Consequently, to preserve hierarchical routing as the topology changes, an entity's address may be required to change—this process is known as "renumbering."

CIDR is an example of the application of hierarchical routing in the Internet, where providers, subscribers, and finally subnets are some of the many different possible levels in the hierarchy. For example, a router within a site need not keep detailed routing information about every possible host in that site. Instead the router maintains routing information on a per subnet basis. Likewise, a router within a provider need not keep detailed routing information about individual subnets within its subscribers. Instead the router could maintain routing information on a per subscriber basis.

It should be noted that there are today a considerable number of "exceptions" to hierarchical routing in the Internet routing system. Such exceptions are routes to destinations whose IP addresses do not reflect the topology of the Internet. A typical example would be an organization with a few dozens (or hundreds) of hosts connected to the Internet, such that IP addresses of these hosts are assigned independent of the Internet Service Provider the organization uses to connect to the Internet.

Further, because the topology of the Internet is not strictly hierarchical, there are a large number of exceptions which will have to be injected into the Internet routing system in the future, in addition to those exceptions which currently exist. Each such exception which is added to the routing system has a deleterious effect on the scalability of the routing system. The number of exceptions which can be tolerated is dependent on the exact technology which is used to support routing. Unbridled injection of such exceptions will certainly cause the Internet routing system to collapse. Similarly, if the sustained rate of injection of exceptions exceeds the rate at which routing technology scales, the Internet routing system is doomed.

### Address ownership and hierarchical routing—mutually unsatisfiable concepts

By definition, address ownership assumes that addresses, once assigned, do not change, and therefore no renumbering can take place. By definition, hierarchical routing assumes that addresses reflect network topology. Therefore, the only way to accommodate both address ownership and hierarchical routing for everyone is to assume that the topology (or at least certain pieces of it) will be permanently fixed. In circumstances where the topology can change arbitrarily, we can accommodate one, but not both—we can either have address ownership, in combination with a non-routable (and therefore non-functional) Internet, or we can have a routable Internet, but without address ownership. In the latter case the concept of "address ownership" is being superseded by a policy of "address leasing."

### Address leasing

An address leasing policy means that an organization acquires its addresses on a "lease" basis. After the lease expires, or if the organization changes its connectivity to the Internet, the organization may either renew the lease (in which case the addresses stay the same), or terminate the old lease and establish a new lease (in which case the addresses change).

*continued on next page*

## Address Ownership Considered Fatal *(continued)*

The leasing policy must be hierarchical as well, so that addresses may be "sub-let" to other organizations. The implication here is that the expiration of a single lease may have effects on organizations which have recursively sub-let a portion of the address space from the main lease.

### Implications on renumbering

Observe that the goal of hierarchical routing in the Internet is not to reduce the total amount of routing information to the theoretically possible minimum, but just to contain the volume of routing information within the limits of technology, price/performance, and human factors. Therefore, organizations which could provide reachability to a sufficiently large fraction of the total destinations in the Internet and could express such reachability through a single IP address prefix could expect that a route with this prefix will be maintained throughout the default-free part of the Internet routing system.

For all other organizations, the reachability information they inject into the Internet routing system would be subject to hierarchical aggregation. Consequently, when such an organization changes its topological attachment to the Internet, but wishes to preserve Internet-wide IP connectivity, the organization eventually needs to renumber to eliminate any exceptional prefixes that they would otherwise inject into the Internet routing system. This applies both to the case where the organization takes its addresses out of its immediate (direct) Internet Service Provider's block and the organization changes its direct provider, and to the case where the organization takes its addresses out of its indirect (one provider away) Internet Service Provider's block, and the organization changes its indirect provider.

If the organization doesn't require Internet-wide IP connectivity, then renumbering can be avoided. In this case the organization may still maintain limited IP connectivity (e.g., with all the other organizations connected to the organization's Internet Service Provider) by limiting the scope of its routing exception to its provider.

Since renumbering is not cost-free, an organization, when presented with a choice of renumbering vs. limited IP connectivity, needs to carefully analyze its own requirements and compare the tradeoffs associated with each alternative (e.g., Application Layer Gateways, Network Address Translation boxes, etc...).

Organizations should be strongly encouraged to deploy tools that facilitate renumbering (e.g,. the Dynamic Host Configuration Protocol [4]). Use of the DNS (rather than `/etc/hosts`) should be strongly encouraged.

### Conclusions

Any address allocation and management policy for IP addresses used for the Internet connectivity must take into account its impact on the scalability of the Internet routing system.

Among all of the possible address allocation and management policies only the ones that yield a scalable routing system are feasible—all other policies are self-destructive in nature, as they lead to a collapse of the Internet routing system, and thereby to the collapse of the Internet.

Within the context of hierarchical routing, address allocation and management policies that assume unrestricted address ownership have an extremely negative impact on the scalability of the Internet routing system.

Such policies are almost certain to exhaust the scalability of the Internet routing system well before we even approach the exhaustion of the current IP (IPv4) address space and certainly well before we will be able to make moderate use of the IP Next Generation (IPv6) [5] address space.

With the current technology or any near-term foreseeable technology, given the Internet's growth rate, the notion that everyone should be able to own IP addresses and receive routing services, regardless of where they connect to the Internet, is technically infeasible. Therefore, maintaining the current status quo (preserving the address ownership concept) becomes fatal to the Internet.

The choices available today are either to make the "address ownership" concept a part of the Internet history and replace it with the concept of "address leasing," or to witness how the Internet, itself, would gradually turn into a part of history.

## Acknowledgements

This article borrows heavily from various postings on various mailing lists. Special thanks to Noel Chiappa, Dennis Ferguson, Eric Fleischman, Geoff Huston, and Jon Postel whose postings were used in this document. Many thanks to Randy Bush, John Curran, and David Conrad for their review and comments.

## References

[1] Kleinrock, L., Farouk, K., "Hierarchical Routing for Large Networks," *Computer Networks,* Volume 1 (1977), North-Holland Publishing Company.

[2] Rekhter, Y., Li, T., "An Architecture for IP Address Allocation with CIDR," RFC 1518, September 1993.

[3] Fuller, V., Li, T., Yu, J., Varadhan, K., "Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy," RFC 1519, September 1993.

[4] Droms, R., "Dynamic Host Configuration Protocol," RFC 1541, October 1993.

[5] Hinden, R. "IP Next Generation Overview," *ConneXions,* Volume 9, No. 3, March 1995.

[6] *ConneXions,* Volume 8, No. 5, May 1994, Special Issue on IP The Next Generation.

Dr. **YAKOV REKHTER** holds an M.S. in Physics from St. Petersburg University (formerly Leningrad University), Russia (formerly the USSR), an M.S. in Computer Science from New York University, and a Ph.D. in Computer Science from Polytechnic University. Between 1984 and 1995 he has been working at T. J. Watson Research Center, IBM Corp. In June 1995 Dr. Rekhter joined cisco Systems. Dr. Rekhter was one of the leading architects, as well as a major software developer, of the Phase II NSFNET Backbone. He contributed to the NSFNET project in the areas of routing, network management, and network topology design. He is one of the leading designers of the Border Gateway Protocol (BGP). He is also one of the leading designers of the Inter-Domain Routeing Protocol (IDRP), ISO 10747. Present activities include work on routing and addressing for IP Next Generation (IPv6), the Routing Arbiter project (under contract with the National Science Foundation), Classless Inter-Domain Routing (CIDR), the Unified Approach to Inter-Domain Routing, host mobility, host autoconfiguration, and IP over ATM. Dr. Rekhter actively participates in many activities of the Internet Engineering Task Force (IETF). Currently he is a member of the Internet Architecture Board (IAB), and a chair of the Inter-Domain Routing Working Group. E-mail: `yakov@cisco.com`

**TONY LI** is currently a Technical Lead for cisco Systems Inc. where he is a bit slinger specializing in IP exterior routing protocols, high performance switching, and fixing really nasty bugs. Tony received his B.S. from Harvey Mudd College and Ph.D. from the University of Southern California. Tony is currently co-chair of the Source Demand Routing, Mobile IP, and CIDR Deployment working groups of the IETF. E-mail: `tli@cisco.com`

# A Firewall Overview

by Ted Doty, Network Systems Corporation

## Introduction

Second thoughts are sometimes a useful thing. Sometimes what seemed like an exceedingly good idea has side effects that you hadn't anticipated, and which need correcting. For example, the dominance of routers (over bridges) in local area networks was caused in part by second thoughts: some of the wonders of interconnecting LANs were pathological events like packet storms. Having thought once about the LAN interconnection problem (bridges), the world thought again (routers).

The Internet itself is going through this right now. The Internet community has striven heroically to develop a technology allowing *universal connectivity.* After essentially complete success, the reaction now (in some quarters, at least) is "What have we done?!" Our second thoughts are that universal connectivity is perhaps not such a hot idea after all, because many of us don't *want* just anyone (especially hackers and other cyberscum) to be able to get to our computers. We have repented of our sins, by installing *firewalls.*

## What is a firewall?

The best definition of a firewall is "...the firewall is there to keep out the jerks." More formally, it is a device that arbitrates access between networks, allowing some types of traffic and blocking other types, based on your organization's policy rules. The rules mostly define which addresses, applications, and users will be considered trustworthy. Data from these sources will be allowed to pass the firewall; all other data will be blocked.

From the point of view of technologies, firewalls tend to come in one of three flavors: *Packet Filters, Circuit Relays,* or *Application Gateways.* All three types of firewalls are typically set up in a similar manner, in that users on the internal network (the network protected by the firewall) have unrestricted access to the outside networks (i.e., the Internet), but users on the outside networks have extremely limited access in past the firewall.

## Packet Filtering routers

Restricting access across networks is not new. Routers have been restricting access based on source/destination addresses and ports since at least the 1980s, with *Packet Filters.* A Packet Filter is an instruction in the router that will typically pass or block a packet based on source/destination address or port number (a note of warning: the capabilities of different vendors' packet filtering capabilities varies enormously; caveat emptor).

There is no rocket science to Packet Filters; filters simply examine particular fields in the datagram to determine things like the sender's or receiver's address, or the port number that TCP or UDP will deliver the data to (this is similar to examining the addresses for routing decisions). Since it is well known where addresses and ports live in datagrams, we can easily extract and compare this information to the rules describing who we are willing to communicate with, and under which circumstances we are willing to do so.

Remember, the Internet does a great job of delivering packets anywhere; my filtering is trying to *break* this, because I may be willing to exchange data with only a particular subset of Internet users.

## Packet Filtering firewalls

Router-based Packet Filters have traditionally allowed filtering based purely on information contained in the packet headers (i.e., addresses, port numbers, time-to-live values, etc.).

Customers have started expecting more from their firewalls over the last year or two. They are no longer willing to settle for restricting access based solely on network address or application protocol used; they are beginning to demand tools that can allow access for particular users, and prohibit the use of particular user commands. In this sense, most router-based packet filtering is not adequate, since most cannot filter application data.

Some firewalls have been designed to specifically address this by filtering the application level data, rather than just network or transport level header fields. Once again, this is not rocket science, since most applications provide the information you want in well-known places (for example, an FTP STOR command always lives in the same place in the packet). Packet Filtering firewalls can allow or block user commands, or logins by particular user names. Several Packet Filtering firewalls provide firewall-to-firewall encryption.

Note that there is nothing that prevents a router from doing this as well. In addition to firewall vendors, there are several router vendors who are beginning to offer this type of functionality.

### Circuit Relay firewalls

A *Circuit Relay* is a firewall that maps incoming connections from a user to an outgoing connection to the user's intended destination, after the user first authenticates herself to the firewall. Everything received on the incoming connection is directly mapped to the outgoing connection, so this provides support for essentially any application you like.

The big advantage of Circuit Relay firewalls is that users must be strongly authenticated (often by hardware based tokens or software based one-time passwords) before being allowed to connect to the destination. Since anyone obnoxious enough to try to hack into your network presumably would have to steal one of these tokens.

A popular free Circuit Relay is SOCKS, widely available on the net.

### Application Gateway firewalls

An *Application Gateway* runs on a UNIX-ish host that examines all the packets sent by particular application programs (for example, FTP). These packets are scanned to determine the commands being sent, and dangerous commands are blocked. Each application is scanned by a program called a *proxy,* so these firewalls are sometimes referred to as *Proxy Firewalls.* Typically, firewalls come with proxies for popular applications, including Telnet, FTP, and SMTP electronic mail; often they also have proxies for new applications like the World-Wide Web.

Each proxy, understands a particular application. As a result, you will see many different proxies running in a firewall. As packets pass through the firewall, the kernel passes them to the appropriate proxy for screening. If a packet passes the proxy's scrutiny, it is sent along its way into the network; otherwise, the packet is discarded and the event is audited (for later examination).

Firewalls using this technology sometimes require an additional user login. Essentially, the user is first forced to authenticate herself to the firewall (via a username/password combination), and then the proxy will open a new connection to the intended destination. The user will have to authenticate herself to the destination as well. As a result, packets do not *flow through* the firewall as they do in a router; fairly substantial translations are performed, as the connection itself is broken and recreated. This imposes a performance penalty on these firewalls.

*continued on next page*

## A Firewall Overview *(continued)*

Most firewalls support one-time passwords for added security, and some are beginning to implement firewall-to-firewall encryption as well. Since only applications for which a proxy exist will be allowed through the firewall, the number of applications allowed to exchange data through the firewall may be limited. For this reason, most Application Gateways provide circuit relay functions as well.

### Firewall design philosophies

Circuit Relay and Application Gateway technologies start from the same minimalist philosophy. Since large programs are much more difficult to demonstrate to be bug-free, or at least relatively bug free, programs are kept as small and as simple as possible (this is the KISS, or "Keep It Simple, Stupid" principle). The operating system itself is modified to remove sections of bug-infested code wherever possible (two examples are *sendmail,* the bloated and buggy UNIX mail program, and the IP Option processing code of the UNIX kernel).

The result is a much less buggy, easier to understand, and therefore safer firewall. However, there has emerged a conflict between the firewall purists who insist on absolute minimalist firewalls, and those trying to add new, high value features like Network Address Translation or encryption. These new features can increase the code size fairly substantially, somewhat "polluting" the minimalist philosophy.

Packet Filtering firewalls take a somewhat different approach, since few filtering implementations use a high level language that lends itself to code analysis. Also, since the packet filters execute in the kernel itself, the firewall itself is vulnerable to a filter bug causing a compromise of the entire system (in theory at least; since most packet filtering devices are not general-purpose computers, this risk is generally considered pretty small in practice). For these reasons, Packet Filtering firewalls tend to rely more on audit ("Black Box") analysis to demonstrate correctness.

### Which firewall is best?

"Best" means different things to different people, so the most we can reasonably do is compare the different technologies. Not surprisingly, each performs differently under different circumstances.

In the past, the common wisdom has held that Application Gateway firewalls are the most secure, and that Packet Filtering firewalls are the least secure (although this is hotly debated in the industry now). The community appears to be taking the approach that a firewall should be either of the Application Gateway type or the Packet Filtering type; Circuit Relays have mostly fallen out of favor, since most users who might be interested in this technology have pretty much adopted Application Gateways (since these also allow the control of user commands).

The two technologies each have their strengths and weaknesses. Packet Filtering firewalls are generally faster than Application Gateways (or Circuit Relays, for that matter). Application Gateways typically support high value features like strong user authentication via one-time passwords. Packet Filtering gateways are typically less expensive, since they can defeat certain attacks that application gateways cannot (such as IP address spoofing attacks, or IP Source Route attacks). Because of their inability to stop these attacks, firewall hosts that use application gateway technology are often installed in a configuration shown below.

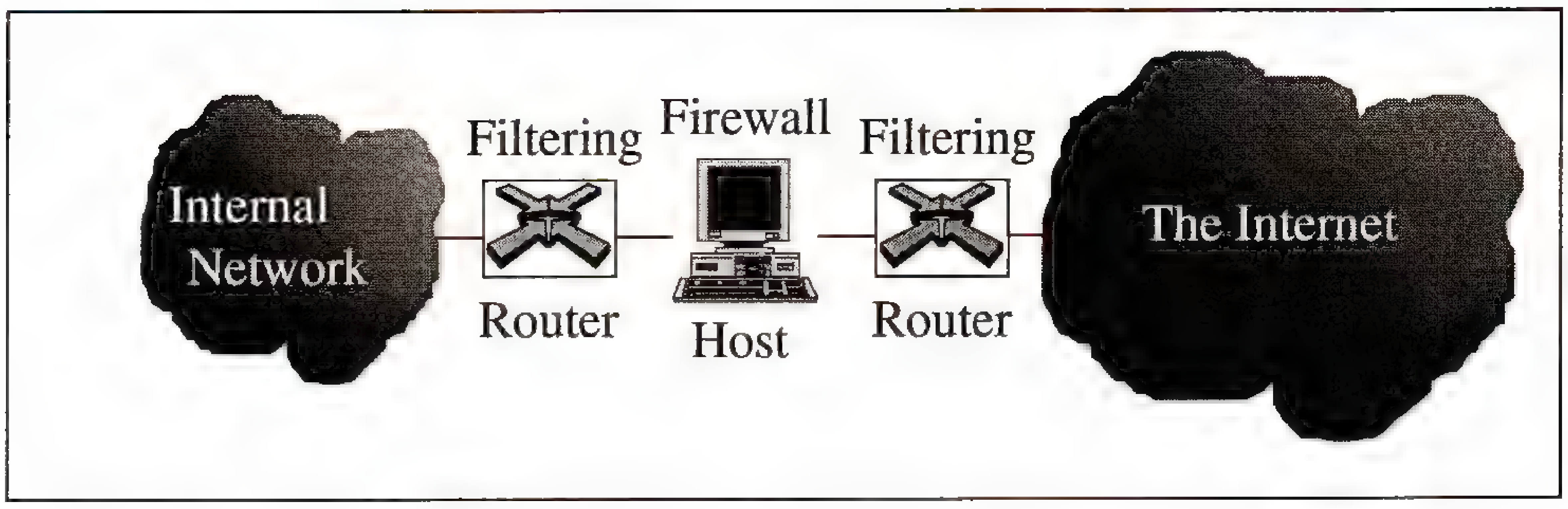

Figure 1: Typical Application Gateway Configuration.

This shows an internal network connected to the Internet using a firewall host and two routers. One router screens attacks from the outside network to the internal one, and the other screens everything sent from the internal network to the outside one. This configuration is called the "Belt and Suspenders" configuration, because you both firewall and filter everything in sight. [A note to readers in the UK: I didn't invent this terminology, I simply report it.] This diagram is commonly seen in firewall literature.

## Conclusions

The firewall market is evolving rapidly, and users have a wide choice of products available. These range from free ("roll-your own") code that you assemble yourself to commercial products costing up to $50,000 or more. A few very broad guidelines may help the beginning of a search for a firewall.

If you have more time than money, you should look at one of the freely available packages. Be warned, however, that Internet security can be frustratingly subtle at times; if you're new to the field, you will have to do a fair amount of research in the literature, or you may expose your network to hackers.

If you choose to buy a commercial firewall, and if you have a low speed connection you want to firewall, or if you don't mind possible performance penalties, you can pretty much shop around for the best price and features. You may very well want to get some expert advise for setting up and running the firewall; most vendors provide this service for an additional fee. If you want to do high-speed firewalling, especially firewalling of departments attached to a corporate backbone, you will probably need to focus on Packet Filtering firewalls.

## Where do I get more information?

One of the most lively groups discussing firewalls is the Firewalls mailing list on the Internet. To subscribe, send e-mail to `majordomo@greatcircle.com`, with the message body consisting of the words "subscribe firewalls." This is a relatively high volume list.

Perhaps the most widely read book on firewalls is *Firewalls and Internet Security*, by Bill Cheswick and Steve Bellovin (ISBN 0-201-63357-4). This is written from the Application Gateway viewpoint, but addresses issues that all firewalls must address.

There are archives of papers discussing many aspects of firewall technology; Greatcircle Associates (`greatcircle.com`) provides a very complete archive available for anonymous FTP; Network Systems provides another (`network.com`).

[Ed.: See also "The Firewall Heresies," in *ConneXions*, Volume 9, No. 6, June 1995.]

**TED DOTY** has been involved in Network Security for ten years. He has worked as an developer, designing and implementing network protocols; as a systems engineer, integrating security products into Local and Wide Area networks; and as a programmer, implementing large scale systems of packet filters. He is now program manager for security products at Network Systems, working on that company's next generation of security products. His e-mail address is `ted.doty@network.com`

# AAUnet: Humble Beginnings of an African Universities' Network

by John Bart-Plange

## Introduction

The *Association of African Universities* (AAU) was founded on November 12, 1967 at a meeting of African university heads held in Rabat, Morocco to, among other objectives:

- Promote interchange, contact and co-operation among university institutions in Africa, and
- Encourage increasing contact between its members and the international academic world.

It was later decided that a permanent Secretariat which will see to the day to day activities towards the attainment of the Association's objectives should be established in Accra, Ghana. The Secretariat is under the executive leadership of a Secretary-General, who is answerable to a 13-member Executive Board that meets once every year to review the performance of the Secretariat. The Executive Board is also answerable to the General Conference, which is the ultimate authority of the Association, and meets once every four years to decide on the broad directions that the Association should take. The General Conference is composed of representatives from all AAU member universities.

With an initial "founding" membership of 34 universities in 11 African countries, AAU membership has grown to encompass 129 universities in 42 countries.

At its Eighth General Conference (which also marked its Silver Jubilee celebrations) held at the University of Ghana, Legon, in January 1993, the delegates adopted a number of resolutions one of which is most relevant for the present discussion:

> "That the AAU Secretariat should intensify efforts at bridging the communications gap between the universities in Africa on the one hand, and the rest of the academic world on the other, as well as amongst the universities themselves."

## Background

Since 1989 AAU, under the able leadership of the incumbent Secretary-General Prof. Donald Ekong, has been pushing for increased interaction and information sharing and exchange amongst its members and with the world's academia as a whole.

In early 1991, as part of a joint *Pan-African Development Information System* (PADIS), Addis Ababa, Ethiopia and the *International Development Research Centre* (IDRC), Canada effort at introducing e-mail into Ghana and some other African countries, AAU Secretariat staff participated in a 3-day training session on Fidonet technology, which was held at the premises of the *Ghana National Scientific and Technology Information Network* (GHASTINET) here in Accra. GHASTINET was designated as node to service the network.

The network did not really get to take off, however. For shortly after the training, none of the designated "points" could connect to the GHASTINET node directly via a modem/telephone line, and "points" had to send messages on diskette by hand to GHASTINET to have them e-mailed, and to receive mail messages by the same means (or, to have them printed and despatched to them, introducing further delays in the process).

One technical difficulty after the other, and then later administrative difficulties, soon made the network inaccessible, and there did not seem to have been enough follow-up support built into the original PADIS/IDRC plan. After a lull of several months, the node was activated again, but still no "points" could connect, and this has been followed by periods of UP-times and DOWN-times.

## AAUNet: A new beginning

The AAU Secretariat's experience with these early networking attempts, as well as our own mandate to bring basic Internet services to our member universities, spurred us on to beat a new path in realising our networking goals. Our experience also taught us the need to undertake a thorough study of the networking environment in Africa as well as world-wide trends, so as to enable us to make wise and forward-looking choices when deciding which technologies to choose in reaching our goal.

It did not take us long to assure ourselves that the TCP/IP Internet was the prize to reach for. But how were we going to get there? What does it take to get there? Do we have the technological means to get there? How many countries in Africa are already there, and how did they get there? These, among other questions, needed to be answered to enable us to see our way clear ahead.

## Turning point

Following the January 1993 General Conference, the AAU in conjunction with the *American Association for the Advancement of Science* (AAAS), Washington, D.C., began preparations for an Electronic Networking Workshop to be held in Accra that same year, as a prelude to the actual take-off of the proposed AAU network. The purpose of the workshop was to help create increased awareness about the benefits of networking. And the strategy was to bring together executive heads and technical personnel from the universities on the one hand, and networking experts on the other, to engage in discussions and exchanges that would help clarify the issues involved, thereby sensitizing and whetting the interest of the universities to co-operate fully with the forth-coming AAU project and, indeed, other plausible networking initiatives.

Come December 1993, and executive heads and technical personnel from seven universities [namely, University of Ghana, Legon, University of Science & Technology, Kumasi, and the University of Cape Coast (Ghana), University of Ibadan (Nigeria), Universite Cheik Anta Diop (Senegal), Universite de Yoaunde (Cameroun), Universite de Cote d'Ivoire (Cote d'Ivoire)] and three networking experts (Lishan Adam from PADIS, Ethiopia, Neil Robinson, University of Zambia, and Mike Lawrie, Manager, UNINET-ZA, South Africa) were engaged in discussions that were to tremendously enhance our knowledge about the Internet—the various means by which one could get connected to it, the cost/performance benefits when compared to, say, fax, etc.

As part of the workshop, an e-mail demonstration was mounted to further clarify what had been said at the lecture sessions. One of the demos which worked then, and which has continued to work up until now, was one mounted by Mike Lawrie—a simple yet powerful DOS-based UUCP system. Having been starved of direct e-mail for so long, AAU pursued this thread of success and, over the past year and a half, the system has proven itself to be most robust—no breakdown that was directly due to it! Besides, the system has the capability to act as mail-router for any number of hosts within the same or even different domains.

*continued on next page*

## AAUnet *(continued)*

Earlier in August 1993, the Internet Society (ISOC) had provided full sponsorship to enable an AAU staff member to participate in the first of ISOC's Training Workshops (which workshops have now become a permanent feature of annual INET meetings) in San Francisco, California. This was to offer an invaluable aid to AAU's plans, as the staff member was exposed to the "blinding light" of the Information Superhighway that had engulfed the technologically advanced world, and was geometrically progressing to reach the rest of the developing world. Participation in the Basic (dial-up UUCP) track, however, ensured that the staff member returned home with something that he could readily work with, given the generally poor (yet expensive) state of the telecommunications infrastructure in Ghana and, indeed, in most other African countries.

### Consultancy

It became clear after the December 1993 workshop that we needed more enlightened advice from networking experts who had had experience working in difficult and shaky developing country telecommunications infrastructure environments as well as having ample knowledge of the full-blown TCP/IP Internet. Having obtained limited funding support from IDRC, Canada, we announced our need for a consultant on the net, and received a goodly number of proposals from all over the globe. In the end one that came from Mr. Dave Wilson, Director of Computing Services, Rhodes University, South Africa, won the day.

The terms of reference of the consultancy was:

- To review existing and planned electronic networking projects in Africa with particular emphasis on identifying African universities participating in such existing and proposed networks, and assessing the effectiveness and reliability of the networks especially in providing to the universities access to basic worldwide electronic network services;
- On the basis of available reports and documents, to review the status of network infrastructure in African countries that have AAU member universities, as well as computer usage in the universities;
- To visit selected African universities to evaluate the information from the reviews, including assessment of needs and infrastructure;
- To prepare for the AAU Secretary-General a report, including recommendations:
  - For the design of a pilot electronic network involving selected universities; and
  - On universities that might be considered for participation in the pilot network, indicating the grounds for selection; and
- To review a draft of the report with the AAU Secretary-General and AAU staff at the AAU secretariat in Accra;
- To conduct a one-week hands-on training and consultative meeting for technical staff of the universities selected for the pilot network.

The universities proposed to participate in the pilot network include: University of Ghana, Legon; University of Ibadan, Nigeria; Universite Cheik Anta Diop, Senegal; Universite de Cote d'Ivoire; Makerere University, Uganda; and Dar es Salaam University, Tanzania.

After visits to the pilot university sites in July–August 1994, Mr. Wilson produced a report that recommended a phased development of the network making use of existing technologies at each stage.

The report emphasises the need to identify and develop local expertise as far as possible, and further advises that responsibility for running each site must lie with local staff and that new developments must be driven by local needs and conditions. While valuable lessons may be learned from previous network startup experience, it should not be necessary to duplicate all the stages passed through by others on the way to full Internet connection in a particular region. The technique of leap-frogging should be used to bypass unnecessary steps in the process leading to this goal, the report says. As the ultimate goal for both the AAU and the universities is full Internet connectivity, the report recommends adherence to Internet standards, albeit even a very small sub-set thereof, from the beginning, so that upgrading can be done continuously as the system evolves without having to heavily reinvest in human and technological resources.

The development of the network as recommended in the report is broken down into 5 phases, with the time-frame for each phase not specified, being dependent on resource and infrastructure availability.

AAU will be happy to share the full report of the consultancy with any interested organizations or individuals.

## Implementation

The AAU Secretariat has already begun to implement the recommendations of the report. Currently there is a multi-user workstation dialup installation in the Secretariat offering mailboxes to individual users in the AAU Secretariat. This has been in successful and reliable operation for over a year. The system is in the process of being upgraded to a multi-user FreeBSD UNIX-based but still dialup net-server offering mail routing with dialup access both via local area networks and to a growing number of institutions and individuals in Ghana.

Already there are six hosts set up on the AAU server. These connect to the AAU server (which is up 24-hours a day) via telephone line and modem, and the number of users at each of these hosts cannot be readily ascertained, but the system is capable of supporting as many users (mailboxes) as a connecting site's PC has room for!

These six sites include the University of Science and Technology (UST), Kumasi Library (they took the software to try to set up their server by themselves but seem to have run into some difficulty, and the AAU System Operator might have to travel to Kumasi to help with the connection), the British Council, Accra office (which connects most regularly), plus three other private individuals and a private company.

It should be noted that the software with which these hosts are set up can enable them to act as hubs themselves, thus producing a rippling effect growth of the network. It is by this means that we would hope, for a start, to provide dial-up Internet access to the university campuses. A national hub at the AAU (and the other key universities participating in the pilot project), will fan out to other universities, and then to the departments, and to the individual student, lecturer, etc. PCs, all using modems and PABX telephone extensions, except where hard-wired LANs already exist, in which case only the LAN server will require the telephone line and modem.

*continued on next page*

## AAUnet *(continued)*

The AAU gateway is ready for any connections, and it is hoped that the University of Ghana, Legon and the University of Cape Coast as well as the two recently opened universities in Ghana would connect shortly.

AAU also expects by the 3rd quarter of 1995 to set up an AAU Information Listserver and Newsgroup which any member university or, indeed, others interested in African higher education can "subscribe" to and can access by e-mail. This will disseminate regularly news and information of interest to member universities including announcements of workshops, seminars, conferences, courses, vacancies, new publications, new developments in the universities, etc.

### Training

In furtherance of the project goals, a pilot training programme for technical personnel from a limited number of member universities is being planned for late July 1995. Its main objectives will be to provide skills in the effective use of e-mail both in its basic and more advanced forms, as well as skills for installing and operating the SNUUPM system (i.e., DOS Version of UUCP, Pegasus Mail and a Newsreader) for dialup mail routing and forwarding.

After the training, the beneficiaries will be expected to carry the knowledge with them to their respective campuses and countries and begin to implement the network. AAU will also provide follow-up support to ensure that the network does not get stalled in any of the sites due to any technical or other difficulties.

Because hard-wired campus-wide high speed networks (e.g., those based on Ethernet) are virtually non-existent on most African university campuses, AAU envisages, as a first step, simple, less expensive networks based on the use of modems on campuses along with PABXs (which seem to be more readily available) to link departments in the universities to the central hubs that would be set up.

With a single UNIX or clone server on each campus, other faculties or departments would connect to it using modems and the PABX telephone line extensions. It is heartening to note that some of the universities participating in the pilot network already have small Ethernet (or similar technology)-based Local Area Networks within some departments and these would form the nucleus from which to develop the full campus-wide networks, as briefly explained above.

This model, it should be stressed, would allow "plug in and go" upgrades to TCP/IP (as the network grows) without having to heavily invest in retraining users. Participants will also be provided with modems to use in starting the nets.

### Technical challenges

As the Ghana wing of AAUnet took off in full swing early April 1995, a number of technical difficulties have surfaced (mainly related to modems), bringing to the fore certain hints that we hope would guide and aid future planners and developers of African networking. These tips, it should be noted, cannot necessarily apply to the whole of Africa; they would apply especially to West and Central Africa, and parts of Eastern, Northern and Southern Africa. Specifically, of course, they apply to Ghana:

- One cannot reliably hope to achieve data speeds above 9600 bps using even 14,400 bps modems when dialing internationally. Above 9600 (or even *at* 9600 to most African countries), the recurrence of errors in data transmission become so rampant as to negate any gains in speed.

The error-correcting software and/or modem keeps re-sending the same packet of data that was in error over and over again until any speed gains are negated, and you invariably spend *more* time on the line sending/receiving the same messages than if you had connected at a lower speed—which connection would have given you reliable, virtually error-free transmissions.

- Connecting to sites in Europe or North America tend to be less troublesome and easier to get than connecting to other African countries—even neighbouring Togo! In either case (but more so in the latter case), however, one does well to allow long time-out periods in both software and modem initialization strings. Time-outs of 60 seconds seem to be the *least* that one can hope to work with connecting anywhere, internationally.
- V.32bis modems with speeds of 14,400 bps as well as V.42bis/MNP5 data compression, V.42 error correction and fall-back/fall-forward capabilities are advised over non-error correcting ones, even though these features may not always work satisfactorily over the shaky phone lines. This notwithstanding, however, the V.32bis modems are more capable than the lower-speed ones in retaining longer dialup connections involving the transfers of large files where the telephone line quality deteriorates in the course of the transmission. The lower-speed non-error correcting modems often just drop/break the connections whereas the error-correcting ones "re-train" the lines and rarely give up on them.
- Ghana P&T indicates that a leased data line of 9.6 Kbps speed would cost in the region of between US$7,000 and US$7,500 per month! Given the almost total lack of knowledge about the Internet in Ghana and, indeed, in most other African countries, our judgment is that dial-up e-mail for at least 6–12 months would be a good "teaser" to create the awareness necessary to make a full TCP/IP connection worthwhile once obtained.

With the interest being shown by individuals and organizations throughout the country, we hope to reach a critical mass within a year or, at most, two. We see the break-even point as the time when the volume of data flow into Ghana (or the other national hubs) and the cost of the phone bills begins to equal at least two-thirds the cost of a leased line that would enable full-blown TCP/IP connectivity.

At the moment, it costs us C2,000 Ghanaian cedis (approx. $2.00) per minute to call South Africa, where we get our mail feed to-and-from the Internet (and Ghana P&T has already hinted about plans for an upward review of prices; local call charges have already been hiked). Within the minute, however, we manage to ship between 18,000 bytes to 25,000 bytes of text. Meaning that our 9600 bps DCE speed (which theoretically should give 960 bytes/second transfer rates) actually gives us transfer rates of between 300 and 416 bytes/second!

Given the excessive overhead data limitations of the UUCP "g" protocol, it is often noticed that the transfer speed improves when few large files are transferred, rather than a large number of small files. The increased use of the AAU gateway should, therefore, see higher throughput speeds.

### Current situation in Ghana

At present (April, 1995), there are at least 4 organizations (including the AAU) purporting to offer dial-up Internet access to the general public.

*continued on next page*

## AAUnet *(continued)*

These include GHASTINET (W._Anim-Dankwa@ghastinet.gn.apc.org), HEALTHNET (DAddo@gha.healthnet.org) and Network Computer System (NCS) Ltd. (support@austin.com.gh).

GHASTINET utilizes Fido technology, whiles HEALTHNET uses a mix of Fido technology and satellite transmissions. AAU uses dial-up UUCP. NCS, a commercial computer company, utilizes dialup TCP/IP (PPP), and have been most aggressive in advertising their service to the public via TV, Radio and Newspapers. None of the other service providers besides NCS does any public advertisements. All these services are offering essentially dial-up e-mail.

There are as yet no clear trends as to the utilization of these services, as all (except HEALTHNET) are very young—3 months old or less. (Please note that this time-frame refers to the more recent attempts at service provision. GHASTINET has been around since 1991, and AAU also have had an internal system running uninterrupted since December 1993. NCS also had an extensive test period of at least one year). At a seminar last December, HEALTHNET reported having connected 30 users.

The **.GH** (Ghana) domain has been registered by NCS, and AAU is in the process of negotiating the administration of the **.GH** domain with NCS, so as to be able to assign hosts under the **.GH** top-level domain to the Ghanaian universities, as well as the indigenously Ghanaian institutions which connect through us.

### Request for support

AAU would appreciate support from donor agencies that would cover the start-up costs of the network on the various university campuses, as well as indicate a commitment to the long-term plans. Essentially and specifically, support is being sought, in the short-term, for the absorbing of telephone bills for at least a period of 6 months to one year, to enable the fledgling networks take root and grow. Also, high-end '486/Pentium computers to act as servers on the university campuses, as well as high-speed modems.

Support with costs of organizing the trainings (involving airfares of trainees, logistics, etc.) would be very much appreciated. Long-term support is also required to enable full implementation of the recommendations of the consultancy report of Mr. David Wilson, to help make campus-wide networks a reality on most African university campuses in the near future. Interested donors and/or well-wishers are welcome to contact the AAU Secretary-General, Professor Donald Ekong <secgen@aau.org>, for full details of our network plans.

### Acknowledgements

Any discussion of AAUnet would be incomplete without special mention being made of certain individuals and organizations who have contributed immensely in one way or the other to our efforts.

- Professor Donald Ekong <secgen@aau.org>: For his vision and exemplary leadership and patience that tolerated the extensive (and, yes, expensive) test phase of the Ghana wing of the network. Many a lesser boss would have given up and not had so much patience with the technicians.

- Mike Lawrie <mlawrie@apies.frd.ac.za>: His venerable back-bench advice and continued support (both technical and other) of our efforts from the very beginning up until now has been most phenomenal—and many an Internaut would bear testimony to his helpfulness. We are greatly indebted to him.

- Dave Wilson <ccdw@kudu.ru.ac.za>: For his expert consultancy advice and continued support of our efforts, and for the free use of the gateway hardware and services at Rhodes University. In this regard, the South African Universities' Network (UNINET) must also come in for special mention for forwarding our mail free of charge across their circuit to the USA, and thence to the rest of the Internet.
- Anthony Kofi Arthiabah <kofi.arthiabah@accra01.x400.gc.ca>: An Internet fan and computer systems engineer with the Canadian High Commission in Accra, Ghana, Kofi is a selfless networker who will devote of his time and resources to see that the AAU node was up and running all the time! He is always only just "a phone call away"! Just inform him of a problem, and he will come over after work (or at times over the week-end), and work with the AAU System Operator late into the night to see that a problem was resolved or at least isolated. The Internet community owes a debt of gratitude to persons such as these who, unfortunately, do not get known for one reason or the other.
- The Internet Society <isoc@isoc.org>: For offering full sponsorship to AAU staff member to participate in the first of its training sessions in 1993 in San Francisco, California, and also now offering to fully sponsor the same staff member to attend the 1995 workshop and conference in Honolulu, Hawaii. We are most grateful.

## References

[1] Mike Lawrie, "Research and Academic Networking in South Africa," *ConneXions,* Volume 5, No. 8, August 1991.

[2] Steve Neighorn, Randy Bush, and Jeff Beadles, "Profile: RAINet," *ConneXions,* Volume 6, No. 5, May 1992.

[3] Bennet, Mark, "Electronic Mail in Zambia," *ConneXions,* Volume 6, No. 9, September 1992.

[4] Bush, Randy and Klensin, John, "Expanding International E-mail Connectivity: Another Look," *ConneXions,* Volume 7, No. 8, August 1993.

[5] Ezigbalike, I. Chukwudozie and Ochuodho, Shem J., "E-Mail for Developing Countries—What They Never Tell You About It," **shem@minster.york.ac.uk.**

[6] Press, L.,"INET '92: The Start of Something Big," *ConneXions,* Volume 6, No. 12, December 1992.

**JOHN BART-PLANGE** holds an Institute of Data Processing Management (IDPM), Sidcup, Kent, UK, Diploma in Data Processing and Computer Programming. In 1991 he was with Masai Developers Ltd., Accra, Ghana (IBM Representative, Ghana) as Course Instructor in computer literacy and DOS and Windows based applications—word processing, spreadsheet and database packages. Since October 1991 he has been with the Association of African Universities (AAU) Secretariat based in Accra, Ghana, where he is responsible for seeing to the smooth running of the PCs, CD-ROM workstations and Macs, as well as the various software that run on them. He is also in charge of the UUCP network node in Ghana as well as the recently set-up Novell NetWare LAN in the AAU Secretariat. He is a member of the Internet Society. E-mail: sysop@aau.org

This publication is distributed on an "as is" basis, without warranty. Neither the publisher nor any contributor shall have any liability to any person or entity with respect to any liability, loss, or damage caused or alleged to be caused, directly or indirectly, by the information contained in *ConneXions—The Interoperability Report®*

CONNEXIONS
303 Vintage Park Drive
Suite 201
Foster City, CA 94404-1138
Phone: 415-578-6900
FAX: 415-525-0194

CONNEXIONS

EDITOR and PUBLISHER Ole J. Jacobsen

Printed on recycled paper